

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 09-026889

(43)Date of publication of application : 28.01.1997

(51)Int.Cl.

G06F 9/46

(21)Application number : 07-177134

(71)Applicant : HITACHI LTD

(22)Date of filing : 13.07.1995

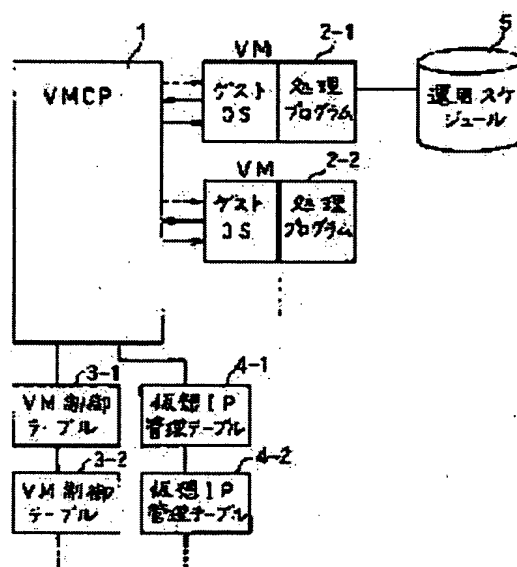
(72)Inventor : YAMAUCHI HIROYUKI
OYAMADA KENICHI
ASAI TAKAYOSHI

(54) VIRTUAL MACHINE SYSTEM

(57)Abstract:

PROBLEM TO BE SOLVED: To change the setting of the assigning amount of the processor for each VM from guests OS working on virtual machines(VM), in the virtual machine system composed of plural virtual machines and a virtual computer control program(VMCP) controlling these VM.

SOLUTION: When the OS on a VM designates a specified VM and issues a processor assignment amount changing instruction, the control is passed to a VMCP 1 and the VMCP 1 changes the processor assigning amount of the VM which is set to a VM control table and is designated to a designated value. Subsequently, the VMCP 1 performs the scheduling in which processor time is assigned to the VM in accordance with the changed processor assigning amount.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平9-26889

(43) 公開日 平成9年(1997) 1月28日

(51) Int.Cl.⁸

G 0 6 F 9/46

識別記号

3 5 0

庁内整理番号

F I

G 0 6 F 9/46

技術表示箇所

3 5 0

審査請求 未請求 請求項の数 1 O L (全 7 頁)

(21) 出願番号

特願平7-177134

(22) 出願日

平成7年(1995) 7月13日

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72) 発明者 山内 宏之

神奈川県横浜市戸塚区戸塚町5030番地 株

式会社日立製作所ソフトウェア開発本部内

(72) 発明者 小山田 健一

神奈川県横浜市戸塚区戸塚町5030番地 株

式会社日立製作所ソフトウェア開発本部内

(72) 発明者 浅井 孝好

神奈川県横浜市戸塚区戸塚町5030番地 株

式会社日立製作所ソフトウェア開発本部内

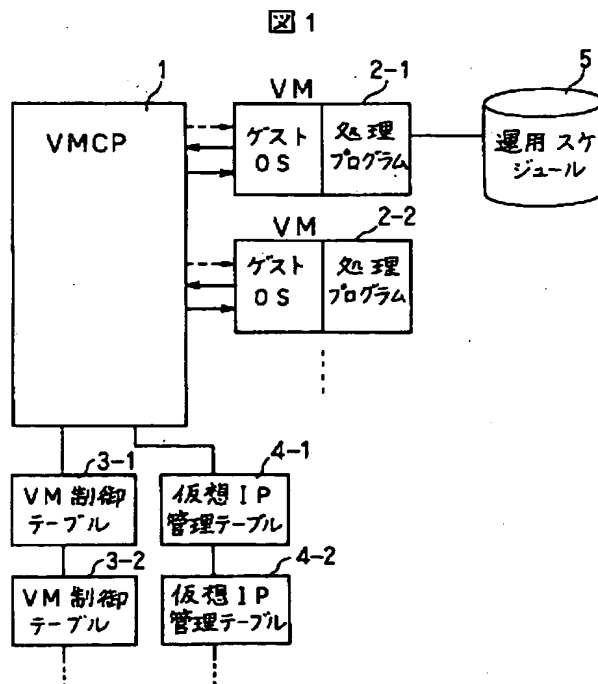
(74) 代理人 弁理士 藤田 利幸

(54) 【発明の名称】 仮想計算機システム

(57) 【要約】

【目的】 複数の仮想計算機 (VM) とこれらVMを制御する仮想計算機制御プログラム (VMCP) とから構成される仮想計算機システムにおいて、各VMに対するプロセッサの割当て量の設定をVM上で稼働するゲストOSから変更可能とする。

【構成】 VM2上のOSが特定のVM2を指定してプロセッサ割当て量変更命令を発行すると、制御はVMCP1に渡り、VMCP1はVM制御テーブル3に設定された指定されたVM2のプロセッサ割当て量を指定された値に変更する。以後VMCP1は変更されたプロセッサ割当て量に従ってプロセッサ時間をVMに割当てるスケジューリングを行う。



【 特許請求の範囲】

【 請求項1 】 複数の仮想計算機 (VM) が設定され、各 VM で動作するオペレーティングシステム (OS) と、各 VM について設定されたプロセッサ割当て量に従って各 VM へのプロセッサ時間の割当てスケジューリングを行う 仮想計算機制御手段 (VMCP) とを有する仮想計算機システムにおいて、

該 OS は外部条件の変化に応じて特定の VM を指定して該プロセッサ割当て量を変更する指令を発行する手段を設け、

該 VMCP は指定された VM の該プロセッサ割当て量を変更する手段を設けたことを特徴とする仮想計算機システム。

【 発明の詳細な説明】

【 0 0 0 1 】

【 産業上の利用分野】 本発明は、仮想計算機システムに係わり、特に各仮想計算機 (VM) に割当てるプロセッサ時間の割当てスケジューリングを制御する仮想計算機システムに関する。

【 0 0 0 2 】

【 従来の技術】 仮想計算機システムは、複数の仮想計算機 (VM) とこれら VM を制御する仮想計算機制御プログラム (VMCP) とから構成される。各 VM では1 つのオペレーティングシステム (OS) が動作する。VMCP と各 VM の OS とは1 台の実計算機の主記憶装置上にロードされ、実行される。VMCP のもつ機能の1 つとして各 VM に対するハードウェア資源としてのプロセッサ時間の割当てスケジューリングがある。VM に対するプロセッサの割当て方式として、プロセッサを特定の VM に占有使用させる占有割当て方式とプロセッサを複数の VM で共用する共用割当て方式とがある。

【 0 0 0 3 】 プロセッサの共用割当て方式では、各 VM についてのプロセッサの割当て量をあらかじめ定義しておき、VMCP はこのプロセッサ割当て量に従ってプロセッサ時間を共用する各 VM に割当てる。

【 0 0 0 4 】

【 発明が解決しようとする課題】 上記のプロセッサ割当て量の定義値を変更する場合には、オペレータが VMCP に対するコマンドを発行して変更しなければならない。このため例えば昼の時間帯と夜の時間帯で VM へのプロセッサ割当て量を変更したい場合、あるいはホットスタンバイシステムにおいて緊急時に現用系から待機系へシステムの切り換えを行った後にただちに待機系及び現用系のプロセッサ割当て量を変更したい場合には、オペレータの介入が必要となり、オペレータの操作ミスを起こしやすく、緊急時の対応が遅れる、システムの自動運転ができない等の問題があった。

【 0 0 0 5 】 本発明は、外部条件の変化に応じて自動的にプロセッサ割当て量を変更する仮想計算機システムを提供することを目的とする。

【 0 0 0 6 】

【 課題を解決するための手段】 本発明は、OS が外部条件の変化に応じて特定の VM を指定してプロセッサ割当て量を変更する指令を発行し、VMCP が指定された VM のプロセッサ割当て量を変更する仮想計算機システムを特徴とする。

【 0 0 0 7 】

【 作用】 運用スケジュールに従ってあるいは緊急事態に応じて関連する VM のプロセッサ割当て量をオペレータの介入なしで変更できる。またプロセッサ割当て量の定義値と実際のプロセッサ使用時間とを比較することにより、プロセッサ使用時間の過不足に応じてプロセッサ割当て量の定義値を変更できる。

【 0 0 0 8 】

【 実施例】 以下、本発明の一実施例について図面を用いて説明する。

【 0 0 0 9 】 仮想計算機システムのハードウェアは、少なくとも1 台の命令プロセッサ (IP) 、主記憶装置、入出力制御装置及び入出力装置で構成される。計算機が複数の IP を有するときには、この複数の IP が共通の主記憶装置を共用し、いわゆる緊密結合のマルチプロセッサの形態で動作する。仮想計算機システムは、複数の仮想計算機 (VM) とこれら VM を制御する仮想計算機制御プログラム (VMCP) とから構成される。各 VM では1 つのオペレーティングシステム (OS) とこの OS によって制御される処理プログラムが動作する。VM 上で動作する OS はゲスト OS と呼ばれる。VMCP 、ゲスト OS 及び処理プログラムは主記憶装置上に格納され、IP によって実行される。各 VM が使用するプロセッサは論理的なものであるから実プロセッサとしての IP と区別するために仮想 IP と呼ばれる。各 VM は少なくとも1 つの仮想 IP を使用し、VM がマルチプロセッサ環境で動作するときには複数の仮想 IP を使用する。VMCP は各 VM について設定された仮想 IP に実際の IP を一時的に割当てる。

【 0 0 1 0 】 図1 は、本実施例の仮想計算機システムの機能的な構成を示す図である。図で点線の矢印は制御の方向を示し、実線の矢印は情報の受け渡しを示す。VMCP 1 は各 VM 2 に IP を割当てるスケジューリングを行う。各 VM 2 を区別するために VM 2 - 1 , 2 - 2 , . . . のように符号をつけている。各 VM 2 ではゲスト OS 及びその処理プログラムが動作する。VM 制御テーブル3 は各 VM 2 を管理するためのテーブルであり、VM 制御テーブル3 - 1 , 3 - 2 , . . . はそれぞれ VM 2 - 1 , 2 - 2 , . . . に対応する。仮想 IP 管理テーブル4 は各仮想 IP に対する実 IP の割当てを管理するためのテーブルであり、仮想 IP 1 , 2 , . . . に対応して仮想 IP 管理テーブル4 - 1 , 4 - 2 , . . . のように符号をつけている。運用スケジュール5 は VM の運用スケジュールについての情報を格納するファイルであ

3

る。運用スケジュール5 は主記憶装置又は外部記憶装置上に格納される。

【0011】VMCP1 はVM制御テーブル3 及び仮想IP 管理テーブル4 を使用して各VM2 にIP を割当てるスケジュールリングを行う。VM上の計算機運転プログラムなどの処理プログラムが運用スケジュール5 を参照して運用のスケジュールを変更するときにはそのゲストOS に指令を発行する。ゲストOS がプロセッサ割当て量変更命令を発行すると制御はVMCP1 に移り、VMCP1 は当該VMに対応するVM制御テーブル3 のプロセッサ割当て量を変更し、以後この変更されたプロセッサ割当て量に従ってVMのスケジュールリングを行う。

【0012】図2 (a) は、各VMに対応するVM制御テーブル3 のうち本発明に関連する項目のデータ形式の例を示す図である。仮想IP 番号3 1 は当VMについて設定された仮想IP の番号である。プロセッサ割当て量3 3 は当VMについて割当てるIP 1 台当りの割当て量を定義するものであり、割当て量はタイムスライスを単位としてその個数で設定される。タイムスライスは通常1 0 ms ~ 2 5 ms 程度の時間である。

【0013】図2 (b) は、各仮想IP に対応する仮想IP 管理テーブル4 のうち本発明に関連する項目のデータ形式の例を示す図である。プロセッサ割当て属性4 1 は当仮想IP が実IP を占有するように割当てるか、他の仮想IP と共用するように割当てるかを示す識別子である。VMが複数の仮想IP を設定するときにはそのプロセッサ割当て属性4 1 はすべて同一である。タイムスライス割当て数4 2 はVMCP1 が当仮想IP にタイムスライスを割当てるときに制御に使用するカウンタである。プロセッサ使用時間4 3 は当仮想IP が実IP を使用した時間の累計値を格納する。実IP 番号4 4 は当仮想IP に対して占有的に又は一時的に割当てられた実IP の番号である。VM制御テーブル3 及び仮想IP 管理テーブル4 は主記憶装置上のVMCP1 の領域内に設定される。

【0014】VMCP1 は、プロセッサ割当て属性4 1 に共用の識別子を設定する仮想IP について、当仮想IP の仮想IP 番号3 1 が設定されたVM制御テーブル3 のプロセッサ割当て量3 3 の値をタイムスライス割当て数4 2 に設定する。次にVMCP1 はタイムスライス割当て数4 2 が1 以上の最初の仮想IP に実IP の最初のタイムスライスを割当て、タイムスライスの満了時に当仮想IP のタイムスライス割当て数4 2 から1 を減じ、タイムスライス割当て数4 2 が1 以上の次の仮想IP に実IP の次のタイムスライスを割当て、タイムスライスの満了時にその仮想IP のタイムスライス割当て数4 2 から1 を減じる。このようにして各仮想IP にラウンドロビン式に順にタイムスライスを割当てていき、すべての仮想IP のタイムスライス割当て数4 2 が0 になった時点で再び上記のようにプロセッサ割当て量3 3 の値を

4

タイムスライス割当て数4 2 に設定してタイムスライスによる各VMへのプロセッサ時間の割当てスケジュールリングを繰り返す。VMCP1 は、仮想IP が実際にIP を使用した時間を計数し、その値を当仮想IP についてのプロセッサ使用時間4 3 に加算する。なおVMが割当てられたタイムスライスを使い切る前に入出力待ち等プロセッサを使用しない状態になると、VMCP1 はそのタイムスライスを打ち切り、次の仮想IP に次のタイムスライスを割当てる。従って各VMについて設定した仮想IP のプロセッサ使用時間4 3 はプロセッサ割当て量3 3 に定義した値を正確に反映したものになるとは限らない。

【0015】図3 は、ゲストOS が発行するプロセッサ割当て量変更命令6 1 とそのパラメータ領域のデータ形式を示す図である。

【0016】図3 (a) は、命令の形式を示すもので、プロセッサ割当て量変更命令6 1 は命令コード、ベースレジスタ (B 1) の指定及びディスプレイスメント (D 1) の指定から成る。B 1 とD 1 とから得られるオペランド領域6 2 には、要求種別コード6 3 及びパラメータ領域アドレス6 4 を設定する。要求種別コード6 3 は命令が要求する機能を指定するもので、(1) プロセッサ割当て量3 3 の変更要求、(2) プロセッサ割当て量3 3 の通知要求及び(3) プロセッサ使用時間4 3 の通知要求の3 種類ある。パラメータ領域アドレス6 4 はパラメータ領域6 5 , 7 0 の先頭アドレスを指定する。オペランド領域6 2 及びパラメータ領域6 5 , 7 0 は主記憶装置上の当ゲストOS の領域内に設定される。

【0017】図3 (b) は要求種別コード6 3 が(1) プロセッサ割当て量3 3 の変更要求又は(2) プロセッサ割当て量3 3 の通知要求の場合のパラメータ領域6 5 のデータ形式を示す図である。パラメータ領域長6 6 はパラメータ領域6 5 の領域長を格納する。VMI D 6 7 は対象とするVMの識別コードである。完了コード6 8 は命令実行が完了したときの完了コードを設定する。プロセッサ割当て量6 9 は要求種別コード6 3 が(1) プロセッサ割当て量3 3 の変更要求の場合には新たに要求するプロセッサ割当て量を格納し、(2) プロセッサ割当て量3 3 の通知要求の場合には当VMのプロセッサ割当て量3 3 を格納する。パラメータ領域長6 6 、VMI D 6 7 及び(1) プロセッサ割当て量3 3 の変更要求の場合のプロセッサ割当て量6 9 はゲストOS によって設定される。完了コード6 8 及び(2) プロセッサ割当て量3 3 の通知要求の場合のプロセッサ割当て量6 9 はVMCP1 によって設定される。

【0018】図3 (c) は、要求種別コード6 3 が(3) プロセッサ使用時間4 3 の通知要求である場合のパラメータ領域7 0 のデータ形式を示す図である。パラメータ領域長6 6 はパラメータ領域7 0 の領域長を格納する。VMI D 6 7 及び完了コード6 8 はパラメータ領

10

20

30

40

50

5

域6 5 の場合と同様である。仮想I P 番号3 1 及びプロセッサ使用時間4 3 はそれぞれ当VMについて設定された仮想I P の番号と実I P の使用時間である。パラメータ領域長6 6 及びVMI D 6 7 はゲストOSによって設定される。完了コード6 8、仮想I P 番号3 1 及びプロセッサ使用時間4 3 はVMCP 1 によって設定される。

【0 0 1 9】図4 は、プロセッサ割当て量変更命令6 1 の処理を行うVMCP 1 の処理の流れを示すフローチャートである。ゲストOSがプロセッサ割当て量変更命令6 1 を発行すると、I P ハードウェアはVMに関する命令であることを検出してVMCP 1 に制御を渡す。VMCP 1 は命令のオペランド領域6 2 にあるパラメータ領域アドレス6 4 からパラメータ領域6 5、7 0 にアクセスし、パラメータを取得する(ステップ1 1)。次にVMI D 6 7 から対象とするVMの識別コードを得て指定されたVMのVM制御テーブル3 を参照し、設定されている仮想I P 番号3 1 から仮想I P 管理テーブル4 を参照し、そのプロセッサ割当て属性4 1 を参照する。プロセッサ割当て属性4 1 が共用であれば(ステップ1 2 共用)、要求種別コード6 3 から要求種別コードを判定する(ステップ1 3)。要求種別コードがプロセッサ割当て量3 3 の変更要求であれば、パラメータ領域6 5 に設定されたプロセッサ割当て量6 9 を指定されたVMのプロセッサ割当て量3 3 に設定する(ステップ1 4)。要求種別コードがプロセッサ割当て量3 3 の通知要求であれば、指定されたVMのプロセッサ割当て量3 3 を当ゲストOSのパラメータ領域6 5 のプロセッサ割当て量6 9 に格納する(ステップ1 5)。要求種別コードがプロセッサ使用時間4 3 の通知要求であれば、指定されたVMのVM制御テーブル3 に設定されている仮想I P 番号3 1 と対応するプロセッサ使用時間4 3 とをパラメータ領域7 0 に格納する(ステップ1 6)。ステップ1 4、1 5 又は1 6 の処理が終わったとき、VMCP 1 は完了コード6 8 に正常コードを格納して(ステップ1 7)、処理を終了する。プロセッサ割当て属性4 1 が占有であれば(ステップ1 2 占有)、完了コード6 8 にエラー終了コードを格納して(ステップ1 8)、処理を終了する。

【0 0 2 0】以下、上記のプロセッサ割当て量変更命令をどのように使用するかについて、ゲストOS及びその制御下の処理プログラムの処理を説明する。ゲストOSがプロセッサ割当て量の通知要求を発行すると、任意のVMのプロセッサ割当て量3 3 を知ることができる。次にゲストOSがプロセッサ使用時間の通知要求を発行すると、指定されたVMについて設定された仮想I P の実際のプロセッサ使用時間を知ることができる。当ゲストOSの制御下の運用プログラムは各VMのプロセッサ割当て量と経過時間とから各VMのプロセッサ使用時間を計算することができる。計算された各VMのプロセッサ使用時間と実際のプロセッサ使用時間とを比較すること

6

によって各VMのプロセッサ割当て量3 3 の設定が妥当であるかどうか判定できる。プロセッサ割当て量3 3 の変更が必要であれば、ゲストOSを介してプロセッサ割当て量の変更要求を発行し、目的とするVMのプロセッサ割当て量3 3 を変更できる。また各VMの実際のプロセッサ使用時間を記憶装置に蓄積すれば、VMの稼働についての統計情報を得ることができる。さらに昼の時間帯と夜の時間帯でVMの稼働状態を変更する場合には、運用プログラムが運用スケジュール5 に従って所定の時刻になったとき関連するVMについてのプロセッサ割当て量を変更することによって目的を達成することができる。例えばVM2 - 1 のプロセッサ割当て量を増加させ、逆にVM2 - 2 のプロセッサ割当て量を減少させるなどである。また例えばVM2 - 1 を現用系、VM2 - 2 を待機系とするホットスタンバイシステムにおいて、ハードウェアの障害又はソフトウェアのバグなどにより、現用系から待機系へシステムを切り換えるとき、システムの切り換え後ただちにVM2 - 1 とVM2 - 2 のプロセッサ割当て量を変更してVM2 - 2 に多くのタイムスライスを割当てるようにVMスケジューリングを行うことができる。なお2 組の実計算機の各々に同一バージョンのVMCP と制御されるVMとを設定し、一方の計算機上のVMを現用系、他方の計算機上のVMを待機系とするホットスタンバイシステムでもVMのプロセッサ割当て量を変更できる。一方の計算機上のVMと他方の計算機上のVMとが通信を行うために、両計算機が通信路又は外部記憶装置を介して接続されていることが必要である。

【0 0 2 1】本実施例によれば、計算機の運用スケジュールに従って目的の時刻に各VMのプロセッサ割当て量をオペレータの介入なしで変更することができる。またホットスタンバイシステムにおいて、現用系のVMから待機系のVMへ切り換えたときただちに各VMのプロセッサ割当て量を変更することができる。

【0 0 2 2】

【発明の効果】本発明によれば、外部条件の変化に応じて関連するVMのプロセッサ割当て量をオペレータの介入なしで変更できるとともに、プロセッサ割当て量の設定値と実際のプロセッサ使用時間とを比較することにより、プロセッサ割当て量の調整を自動的に行うことができる。

【図面の簡単な説明】

【図1】実施例の仮想計算機システムの構成を示す図である。

【図2】実施例のVM制御テーブル3 及び仮想I P 管理テーブル4 のデータ形式を示す図である。

【図3】実施例のプロセッサ割当て量変更命令とそのパラメータ領域のデータ形式を示す図である。

【図4】プロセッサ割当て量変更命令の処理を行うVMCP 1 の処理の流れを示すフローチャートである。

50

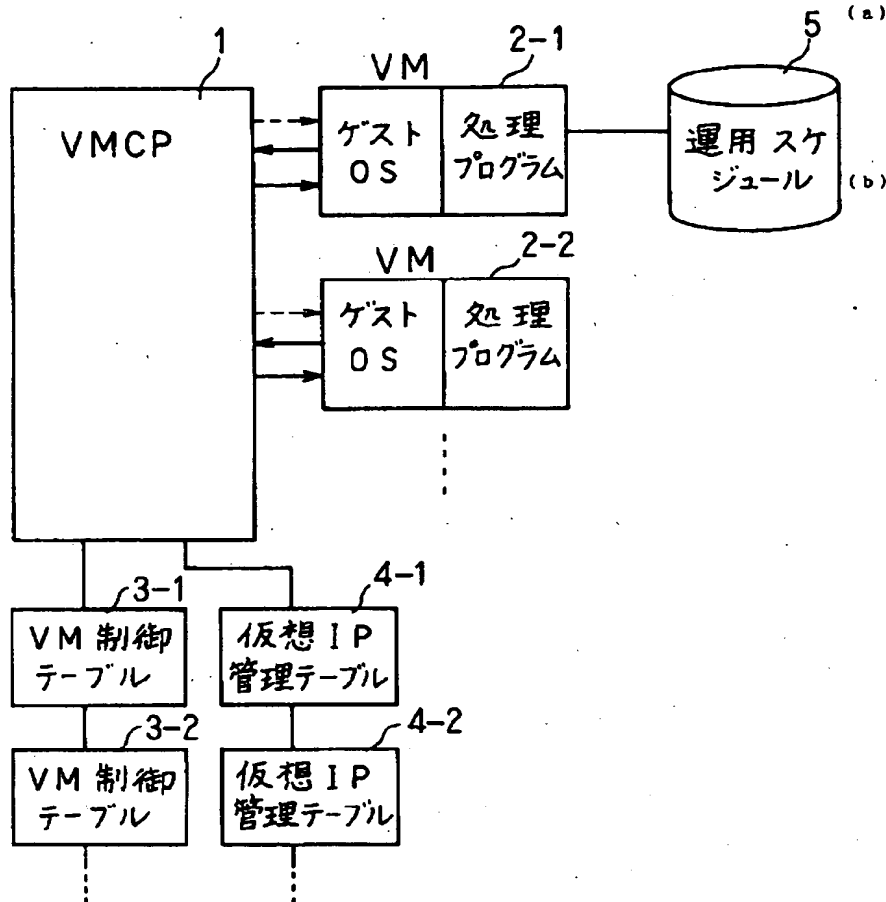
【 符号の説明 】

1 : VMCP、2 : VM、3 3 : プロセッサ割当て量、

6 1 : プロセッサ割当て量変更命令、6 9 : プロセッサ割当て量

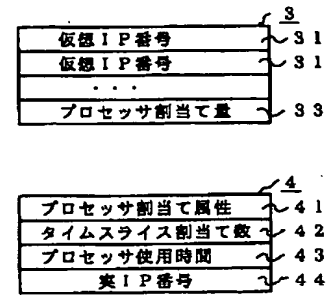
【 図1 】

図 1



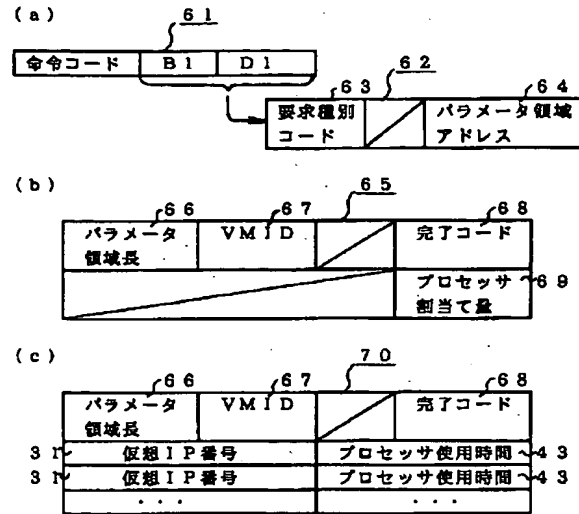
【 図2 】

図 2



【 図3 】

図 3



【 図4 】

図 4

